PC Cross-Zap

A File Transfer & Disk Zap Program
for
IBM PC & compatible computers

PO Box 51155, Raleigh, NC 27609

READ THIS FIRST

Notice: This program is capable of writing to your floppy disks. It is therefore possible to overwrite and lose valuable data if you are not careful. Please be sure to retain backup copies of all disks considered irreplaceable.

Please read, and make sure you thoroughly understand this manual. No responsibility can be accepted for damaged or lost data through misuse or mistakes on the part of the user.

Every effort has been made to ensure this program operates as described in this manual. If you find any problem, please report it directly to Hypersoft, and we will exert our best efforts to resolve it and return you a revised copy.

Trade Marks:

VERY IMPORTANT NOTICE

Due to the way many TRS-80 Operating Systems format disks, it is quite common for the first sector on the disk to occur too soon after the index hole for an IBM computer to read it properly. If this happens you will get a "Sector Not Found" error message when PC Cross-Zap tries to read such a disk. To prevent this, simply take a small gummed label or disk write protect tab and stick it on the disk jacket so it covers the index hole aperture (the small circular hole about 3/16" diameter about 1" South-East of the disk center). This will stop the PC floppy disk controller from resetting when the index hole comes round. Remove this if you intend to use the disk in a TRS-80 or you need to reformat it.

PC Cross-Zap

Manual V1.2 03/05/87

Introduction.

PC Cross-Zap (or PC-XZ for short) is a program for owners of IBM-PC and compatible computers that allows users to access the files on floppy disks made on TRS-80 computers. Features of PC-XZ include:

o File Transfer

Allowing copying of files from TRS-80 to PC or from PC to TRS-80 type floppy disks. Files can be copied as-is (Image mode) or converted on the fly (ASCII and Basic modes). In the Basic transfer mode, TRS-80 Basic programs are converted to a form compatible with BASICA or GWBASIC.

o Disk Format.

Using PC-XZ you can format a disk that will then be readable on your TRS-80 by your favorite DOS. All double density DOS formats are supported including TRSDOS 2.7/8, TRSDOS 1.3, DosPlus 3.4 and 3.5, LDOS 5.1, NewDOS-80 V2.0, MultiDOS, TRSDOS 6.2. Double Sided and 35, 40 and 80 track operation are supported as appropriate. Since PC hardware will not handle single density only double density formats are supported. Several Model I operating systems support double density through the use of a single density track zero. PC-XZ supports this format to the extent that you can read and write files to such disks as long as the files are not on track 0. You cannot format disks like this however.

o Disk Directory.

This will display either a short form or a full directory of your TRS-80 disk. A short form directory is also available for PC/MS-DOS disks.

o Disk Navigator.

In Disk Navigator mode you can examine, modify or repair disk sectors of all types of disks including TRS-80 and MS-DOS disks. Special features are provided to identify the meanings of bytes in directory entries, GAT, HIT and FAT sectors. An option to automatically repair damaged GAT and HIT sectors is also provided.

o Shell Command.

The Shell Command allows you to temporarily to leave PC Cross-Zap and execute any DOS command or other PC program providing you have enough memory.

o Save Configuration.

The Save Configuration Command allows you to save the current default settings, including Alien format specification for automatic restart next time the program is used.

1.0 Hardware Requirements.

PC Cross-Zap will run on any IBM-PC or compatible computer that has the following minimum features:

- o Standard PC or MS-DOS 2.0 or higher operating system.
- o Minimum of 256 Kilobytes of memory.
- o At least one 40 track 5.25" floppy disk drive.
- o Standard floppy disk controller.
- o DMA controller. (Model 1000 owners please note.)
- o Monochrome or Color Graphics Adapter.

The program will run more smoothly with two drives or at least one drive and a hard disk. A RAM disk can be used if you have sufficient standard, extended or EMS memory.
You need 256k of memory for all functions except the Shell Command which needs at least 384 Kilobytes for useful operation. If you run any memory resident programs simultaneously with DOS and PC Cross-Zap you will need proportionately more memory.

Versions 1.0F and later support High Density floppy disk drives of the type found on AT type computers like the Tandy Model 3000. These drives are optional equipment on most AT's. If you have one you can read, write and format 40 or 80 track TRS-80 type floppy disks. It is not generally advisable, however, to write to floppy disks on an 80 track drive if the disk was originally formatted on a 40 track drive.

PC Cross-Zap also supports regular 80 track drives which you can add to your PC just like the standard equipment 40 track drives. PC-XZ will automatically double step if you select a 40 track TRS-80 format in an 80 track drive.

The program has been tested and found to run on all of the following: Standard IBM PC, XT and AT, Tandy Models 1000 (with DMA), 1000EX, 1000SX, 1200, 3000, and a number of other compatibles. In fact it should run on all compatibles.

2.0 Getting Started.

PC Cross-Zap is supplied as a single file XZ.EXE on a floppy disk. The first thing you should do is make a backup and put the original away in a safe place.

You may copy the program to a disk with an operating system on or to your hard-disk if you have one. To run the program, simply type:

XZ <Enter>

If you are running it from a floppy disk you can remove the disk containing PC-XZ freeing your drives for file transfer. If you have an AT with High Density (1.2 Meg) drives see section 3.8 on specifying floppy drive types.

If you are running PC-XZ for the first time, you may want to set up a configuration file. This is done by making various menu selections and then, from the main menu selecting number 8 from the main menu. This will create a file called XZCONFIG.DAT in your current disk directory.

When you run PC Cross-Zap it will display choices in the form of menus. For the most part you can select the action you want by a single key stroke as indicated in the menu. The Escape Key, <Esc> in the menus will take you to the next higher menu or to DOS if you are in the Main Menu.

Where the function you select requires a typed answer it may offer you a default as shown in parenthesis (). To select the default simply press the Enter Key on your keyboard.

Before we proceed, it would be helpful if the Reader understands the meanings of various terms used in this manual. Please skip this section if you think you know all this information. If you find something later that you don't understand return here to see if it is explained.

The Menus and the Keyboard.

All the Menus display a list of options, either numbered or lettered. To select an option simply press the appropriate number or letter key.

The function keys F1 to F10 are also used for various special operations so where you see a reference to say F10 this means the key with F10 printed on it.

The Escape key referred to in the menus as <Esc> is usually marked Esc. The Return or Enter Key may be marked with a name such as 'Retrn' or a graphic symbol such as an arrow which points down and left.

Memory and Disk Terminology

o ASCII The American Standard Code for the Interchange of Information. Used by computers to store text files. Each character and

printable symbol plus other needed control characters is assigned a numerical code in the range 0 to 255.

- o Bit — The smallest unit of memory storage equivalent to one cell in a memory chip. The cell can take one of two possible states: either 'off' or 'on' represented numerically as 0 or 1.
- o Byte — A unit of memory storage capacity. One Byte is 8 bits and can store one text character.
- o Nibble — A unit of memory storage equal to half a byte or 4 bits.
- o Hex — Hexadecimal (base 16) numbering system is used by computers and users to represent programs and data. Convenient because an 8 bit byte can be represented by just 2 hexadecimal digits. Hexadecimal counts 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12 etc. The largest 8 bit number is FF hex or 255 decimal. Hex numbers are usually shown with an H appended. E.g. 10H would be 16 decimal.
- o Track — The path traced on a floppy disk by the recording head as the disk spins beneath it. A typical disk has 40 tracks. Each track can hold up to 6250 bytes of recorded information using double density recording. Tracks are numbered from 0 so the highest track on a 40 track disk is number 39.
- o Sector — A segment of a track. The size of a sector is measured in bytes. Not all the track is available for useful information. Gaps are recorded between the sectors with special control and sector label information. Each sector on the disk is identified by its own unique label. Typical sector sizes are 512 bytes for MS-DOS and 256 bytes for TRS-80 type DOSs.
- o Side — Disks have two physical sides both of which may be recorded upon. Most MS-DOS systems use both sides. Until the advent of the TRS-80 Model 4D all official TRS-80 systems used single sided formats. Many third party operating systems provide the capability of double sided operation. Side 0 is the back (opposite the label), side 1 is the front.
- o Granule — Or Gran for short. A TRS-80 unit of disk storage. Typical Gran sizes are 3, 5 or 6 sectors. Allocation of space on disks is by Grans so the smallest amount of space occupied by a file is 1 Gran. Example: on a TRSDOS 1.3 disk which is single sided and has 18 sectors per track numbered 1-18, the granule size is 3. Granule 0 would be track 0 sectors 1-3, Gran 1 would be sectors 4-6 etc.
- o Cluster — The MS-DOS unit of disk space. On floppy disks a cluster is 1 sector on single sided disks and 2 sectors on double sided disks. On hard disks a cluster can be as large as 16 sectors (8192 bytes). Cluster number 2 is always the first cluster after the directory.
- o GAT — Granule Allocation Table. A map of disk space kept by TRSDOS to speed finding space to put new or expand existing files. Space is measured in Grans.
- o HIT — Hash Index Table. A table kept by TRSDOS that speeds finding whether a file of a given name exists in the directory. Each filename is encoded or 'Hashed' into a one byte code called a Hashcode.
- o FAT — File Allocation Table. A Map of disk space kept by MS-DOS to speed finding space to put new or expand existing files. Space is measured in Clusters.

3.0 The Main Menu.

The Main Menu is your key to selecting all the functions and sub-menus of PC Cross-Zap. It appears when you first start the program after you have chosen an alien disk format. If you are in any sub-menu you can return to the Main Menu by pressing the Escape key. To leave PC-XZ and return to DOS simply press Escape again.

Following is a detailed description of the Main Menu functions and options.

3.1 Select Format. (press key '1')

This allows you to select a new alien disk format. Only one alien disk format at a time may be active. For the purposes of the Disk Navigator feature (see later) MS/PC-DOSs are also included in the list. Whenever you select an operation that works on an alien disk the chosen format will be assumed. If the disk is not of the assumed format then you will either get an error message if the disk is unreadable or you may corrupt the disk if you try and write to it. It is your responsibility as the user to make sure that the disk is of the correct format. If you are not sure try displaying the directory or looking at it with Disk Navigator before proceeding to any other operation. Attempting to read a disk will not harm it, even if it is of the wrong format.

The Select Format option will display a screen of alternative alien disk formats. Each has a code number or letter by the side of it. To choose your format type the appropriate number or letter. When you do so a window will open displaying your choice and asking if you want to proceed. For example, if you have selected TRSDOS 6.2 single sided format, the following should be shown:

```
Format Selected: Model IV TRSDOS 6.x    ss
   Y if OK,  F7 for info,  F8 to revise.
```

Pressing any key but Y, F7 or F8 will return you to the format selection. Pressing Y for Yes will activate the format and return you to the Main Menu. Pressing F7 will open a window which will show you details of the format such as number of tracks, sides, sectors, directory track etc.

The format in each case defaults to the standard for that DOS however the F8 option allows you to modify the directory track number and the number of tracks. For instance the LDOS standard is 40 tracks with the directory on track 20 and you might want to change these to 35 and 17 respectively.

3.2 File Transfer. (Press '2')

Pressing the '2' key brings up the File Transfer sub-menu. This has a range of options allowing copying of files to and from alien disks.

3.2.1 Copying Alien to PC

To copy files from a TRS-80 type floppy disk to an MS-DOS disk use this option. Shown on the screen will be the default disk drives. Unless otherwise specified, files will be copied from the disk in the default alien drive to the disk in the default MS-DOS drive using the default path (MS-DOS subdirectory) as shown.

When the Copy Alien to PC option is chosen you will be prompted for a filespec thus:

Enter Filespec: [drive:]filename/ext [-option]

Items in square brackets [ ] can be omitted in your reply. For example, you can enter the following forms of filespec:

filename/ext	e.g. test/bas
drive:filename/ext	e.g. b:patient/dat
drive:filename/ext -opt	e.g. b:postage -i

Wildcards are permitted in filenames, for instance a:*/* would cause the copying of all files on the disk in drive A. a:xy*/bas would copy all files who's names begin with xy and whose extensions are /BAS. Note all characters are converted to upper case inside PC-XZ.

There are three possible modes when copying from TRS-80 to MS-DOS. These modes are: ASCII, Basic and Image. They can be selected by appending a -A, -B or -I to the filename as shown in the example above.

ASCII transfer mode converts TRS-80 text files to MS-DOS format, appending a line feed (0AH) to each carriage return (0DH), converting multiple spaces to TAB characters where possible, converting the [character to ^ and appending a 1AH end of file mark. The MSB (most significant bit) of each byte is masked out. The file length is also correctly recorded in the directory.

BASIC transfer mode converts standard Compressed Format TRS-80 BASIC files to Compressed Format MS-DOS files readable by BASICA or GWBASIC. Spaces are inserted where necessary and the syntax is adjusted including replacing PRINT @ statements with LOCATE X,Y. Certain other statements that have no direct counterpart in PC Basics or that might give trouble are flagged by tacking on a ! character. This forces BASIC to give an error message and tell you where it occurred so that you can quickly find and correct the line. In addition a :'@#$% statement is appended to each line with one or more potential errors.

Image Transfer mode transfers files as-is without any modification. Use this for Binary files, Data files etc.

3.2.2 Copy PC to Alien

Again, unless otherwise specified, files will be copied from the default drive and path (subdirectory) of the MS-DOS system to the default TRS-80 disk as shown in the defaults at the bottom of the File Transfer Menu screen. When you select the Copy PC to Alien option you will be prompted for the files to copy thus:

```
Drive:\Path\Spec
```

At this point you can reply with any standard MS-DOS filespec. Wildcards are allowed and the Drive and Path are optional. If you omit either, the defaults will be used.

Note that the filenames in MS-DOS remain the same when the file gets copied to TRSDOS except that the period (.), separating the body of the name & the extension, gets replaced by a slash (/). For example MYFILE.BAS becomes MYFILE/BAS. A problem can arise because some MS-DOS filenames are not legal in TRS-DOS. In TRS-DOS only the characters 0-9 and A-Z are permitted whereas MS-DOS also allows other symbols such as '-' in the name. TRS-DOS names must start with a letter so such names as 66/DAT are illegal. Any files with such names as this must be renamed: e.g. make 66.DAT into F66.DAT before copying. See use of the SHELL command.

Valid copy modes are ASCII and Image. Basic files are not converted in the PC to TRS-80 direction. ASCII files are converted in the reverse manner to that described in Section 3.2.1 above. TAB characters are expanded to spaces.

3.2.3 List Alien File.

This option allows you to read text files on your TRS-80 disks without copying them. To call this option press the '3' key and you will be prompted for a filename. At this point you should enter a valid name of a file on the disk in the default TRS-80 drive or you may include a drive letter as well. Valid responses are, for example:

```
myfile/asc
b:maillist/dat
```

The file will list to the screen of your PC. You may suspend the listing by pressing any key. To resume listing press the Enter Key or to abort press the Escape Key.

3.2.4 Display Directory of Alien Disk

This will display a short form directory of the alien disk in the alien default drive. There are no options.

3.2.5 Display Directory of MS-DOS disk.

As 3.2.4 above but for the MS-DOS default disk.

3.2.6 Change Defaults

The current defaults for alien and MS-DOS drives are shown at the bottom of the File Transfer Menu Screen. To change these press '6' and you will be asked, item by item, for replacements. As each question is asked you will be given the current value, shown in parenthesis. To accept this without revision press the 'Enter' key.

==========

3.3 (Main Menu) Directory

Pressing '3' at the Main Menu gives a short-form directory of the alien drive just like the File Transfer Menu. However you can also type

DIR or DIR -a

DIR gives a detailed directory listing showing file size, in Records and Grans, End Of File, File Date and other attributes. Only visible files are shown.

DIR -a shows all files including System and Invisible types.

==========

3.4 (Main Menu) Format Alien Disk.

This option allows you to prepare a new data disk for your TRS-80 type DOS. The disk will be formatted in the currently selected alien format. All alien formats can be used except for the fact that some TRS-80 Model I formats have a single density track 0. Since the floppy disk controller on the PC will not read, write or format in single density you will not be able to make such disks on your PC that will be acceptable to your Model I DOS. If you try and format any of these formats (marked in the menu with a *) then track 0 will be formatted in double density. You can still use these disks to make backups and repairs of your old Model I disks providing you only ever intend to read them on your PC under PC Cross-Zap.

To format a disk, press '4' at the Main Menu and then follow the questions listed below. Note pressing Enter at any question means 'Use the Default shown'.

Volume Name: The default is DATADISK. Enter any suitable name, 8 characters maximum or press 'Enter' to accept the default name 'DATADISK'.

Volume Date: The default is the current MS-DOS system date. Press 'Enter' to accept this or type the date in the format MM/DD/YY.

Number of Tracks: The default is the number of tracks defined in the current alien format selected.

Drive to format: The default is always A. Press 'Enter' to accept this or any letter corresponding to an existing valid floppy disk drive. High density drives are NOT supported. Most PCs have one or two drives so the answer should normally be 'A' or 'B'.

Insert disk in drive and press any key when ready (or <Esc>)
At this point, pressing 'Escape' will abort the format and return you to the Main Menu. Pressing any other key will start formatting the disk. Note that the program does not check that the disk is blank so it is your responsibility

to check this before proceeding. Failure to do so could result in loss of valuable data or programs.

Once formatting has started it will proceed to completion unless an error is encountered. The only way to abort the formatting procedure is to open the disk drive door which will cause an error message to be displayed. Select 'A' to abort.

Your completed disk will be a data disk with BOOT/SYS and DIR/SYS files only. Unlike many disks made on TRS-80s it will not need a write protect tab covering the index hole. If you intend to move data back and forth on a regular basis it is suggested that you do it in a disk formatted on PC Cross-Zap.

==========

3.5 Disk Copy (Key '5' on Main Menu)

This provides a sector by sector copy of a whole disk. It requires the destination disk to be preformatted. It is slow and is not recommend for general purpose disk copying.

To use, press '5' at the Main Menu. You will be asked for source and destination disk drives. They may be the same although this is even slower since the transfer buffer is only 16 kBytes & therefore you will have to swap disks many times.

Disks Formatted and Copied using PC Cross-Zap will not need the index hole covering.

==========

3.6 Disk Navigator

Disk Navigator is a major subsection of PC Cross-Zap, designed to allow the user to read, modify and write sectors of floppy disks in any format listed in the Format Selection Menu. This is particularly useful if you no longer have access to your TRS-80 and have problem disks.

Special attention has been paid to providing the user with guidance when viewing sector data. See Chapter 4 of this manual for full details.

==========

3.7 Shell Command (Key '7' on Main Menu)

The Shell Command allows you to temporarily leave PC Cross-Zap and execute any series of DOS commands after which you can return to the program in exactly the same state as you left. The Shell command causes DOS

to run another copy of COMMAND.COM while leaving PC Cross-Zap in memory in suspended animation. When you are ready to return from DOS simply type the command 'EXIT'.

Note that the Shell Command is not fully supported in releases of MS and PC-DOS earlier than 3.0. As a result, while it will work in PC Cross-Zap, you will have to reboot your computer when you finally return to DOS.

To use the Shell Command, press '7' at the Main Menu and you will be given a prompt. At this point you can type a single line DOS command such as DIR C:/W which will execute and return immediately. ALternatively, if you want to execute a series of DOS commands, press the 'Enter' key at the Shell prompt. The screen will clear and you will be in DOS until you type 'EXIT' to return.

Examples of use of the Shell Command include:

- Renaming Files
- Change Subdirectories
- Copying MS-DOS files
- Listing or Printing MS-DOS files
- Determining free disk space.

==========

3.8 Save Defaults (Key '8' on Main Menu)

If you select this option then PC Cross-Zap will create a file called XZCONFIG.DAT on your MS-DOS disk. This file contains information about the configuration of your system, the default drives you have selected and the current Alien Format selected. Each time you run PC-XZ it looks for this file and, if it finds it, the information contained will be loaded automatically, saving you the trouble of reconfiguring the program every time you run it.

You can, from the main menu, tell PC-XZ what kind of drives you have. To do this type: D=X where D is a valid floppy drive letter and X is 40 or 80 for regular 40 or 80 track floppy disk drives or X is hd for a high density drive. Use X=0 to remove a specification altogether. For example, b=40 sets drive B: to be a 40 track drive, a=hd sets drive A: to be a high density AT type drive.

If you have an AT type computer, PC-XZ automatically analyzes the equipment status registers and determines the number and type of floppy drives.

==========

4.0 The Disk Navigator

Disk Navigator is essential for examining, modifying and repairing disk sectors. The Disk Navigator Menu gives the following options:

1) View/Edit Disk Sectors
2) View/Edit Directory Sectors
3) View/Edit FAT/GAT sectors
4) Fix Bad GAT sector
5) Fix Bad HIT sector
6) Change Default Drive.

4.1 View/Edit Disk Sectors (Press '1')

This allows you to view the data in a specific sector. To use, make sure your disk is in the default drive (or change the default as necessary using option 6.) and press '1' at the Disk Navigator Menu. You will be prompted:

Enter Track, Side, Sector:

Reply (for example) 1,0,3<Enter> to read sector 3 on side 0 (back) of track 1. Note tracks are numbered from 0, side 0 is the back, side 1 is the front.

Once you have made your selection the sector will be read into memory and displayed. A flashing cursor points to the byte under examination. Use the arrow keys to move it about. Lower right is the Relative Byte Indicator that shows the position of the cursor relative to the start of the sector.

In sector examine mode the following keys are active:

Up Arrow:	Moves cursor up one line (-16 bytes)
Down Arrow:	Moves cursor down one line (+16 bytes)
Right Arrow:	Moves cursor right 1 char (+ 1/2 byte)
Left Arrow:	Moves cursor left 1 char (- 1/2 byte)
Home Key:	Moves cursor to top left hand corner
End Key:	Moves cursor to lower right hand corner
PageUp or ;	Move up in memory 256 bytes. If end of Sector exceeded, read next sector.
Pagedn or -	Move down 256 bytes. Read next sector if top of screen exceeded.
Shft lft-arrow	Read same sector, Side 0
Shft rgt-arrow	Read same sector, Side 1
Shft up-arrow	Read next higher sector
Shft dwn-arrow	Read next lower sector
Shft PageUp	Get 1st sector of next higher track
Shft PageDn	Get 1st sector of next lower track
T	Select New Track, Side, Sector
M	Turn on Modify Mode

Z	Zap - Write sector to disk.
Esc	Return to Disk Navigator Menu.

Once Modify Mode is selected you can change sector data in either Hex or ASCII modes. In Hex mode you type any of the keys 0-9 or A-F and modify the bytes a nibble at a time. In ASCII mode you type any printable key character and the byte equivalent to that is placed at the current cursor location.

In Modify mode the following special keys are active:

Function Key F2	Modify Mode OFF
Function Key F3	Switch to Hex Modify Mode
Function Key F4	Switch to ASCII Modify Mode.

While in modify mode you are changing only the copy of the sector data in computer memory. To modify the data on disk, select F2 to exit modify mode and then Z to write the sector back to disk.

==========

4.2 View/Edit Directory Sectors.

This is identical to View/Edit Disk Sectors except for the following:

1) By choosing this option PC Cross-Zap automatically selects the correct side, track and sector for the start of the directory.

2) Displayed below the sector is specific information about the byte under examination. Each directory sector takes (typically) 32 bytes. The number of the entry and the byte offset within the entry are shown on the lower left below the directory sector number.

3) The following special keys may be active in View/Edit Directory Sectors:

F7	Show Disk Format Information
F8	Select New Track, Side, Sector
F9	Return from Viewing Extent/Cluster
F10	View 1st sector of Extent or Cluster

When the cursor is on an Extent Descriptor pair within a directory entry, pressing F10 will take you to the track, side and starting sector of the relevant extent or cluster. Pressing F9 will return to the same place you left from.

==========

4.3 View/Edit FAT/GAT

Like the View/Edit Directory option this will take you to the point where you are viewing the first sector of the FAT (MS-DOS) or GAT (TRS-DOS). In the case of TRS-80 type DOSs, pressing PageUp will take you up one sector so you are looking at the HIT sector.

When viewing the GAT sector of a TRS-80 disk the following may be observed:

1) The allocation bytes are identified and the logical cylinder number is given. A table is displayed showing which granules are used and which are free.

2) A similar table is available when the cursor is over the bytes corresponding to the track/sector lockout table.

3) Bytes corresponding to the disk name and date are identified as are those for the AUTO command.

When viewing the HIT sector of a TRS-80 disk:

1) Each byte corresponds to a directory entry. The byte is the hashcode of the filename. The location of the byte in the table is determined by the directory sector number and the entry number within that sector. Three different formulae apply according to whether the DOS is TRSDOS 2.7/8, TRSDOS 1.3, or any other DOS. The correct formula is used and the Directory sector and Entry are given.

2) Pressing the F10 key will take you immediately to view the corresponding directory entry. While there pressing F9 will return to the same location in the HIT sector.

When viewing the FAT (File Allocation Table) of an MS-DOS disk:

1) Each 3 byte group describes the allocation or use of two clusters-worth of space on the disk. Because of the scrambled way this is done (see the Chapters on disk formats) PC Cross-Zap decodes these for you. Wherever the cursor is, corresponds to one of 3 bytes in a 3 byte group. Those 3 bytes are highlighted in two colors so that you can see which parts of which bytes are used to describe the two clusters. See Chapter 6 on MS-DOS formats for more information.

2) Press F5 for 1st sector of the 1st cluster.
Press F6 for 2nd sector of the 2nd cluster.

3) When viewing either cluster (having pressed F5 or F6) pressing F9 will return you to the same place in the FAT that you left from.

5.0 TRS-80 Disk Formats

This chapter gives details of how data is organized and recorded on TRS-80 type disks. A knowledge of this information is essential if you are going to use the Disk Navigator feature of PC Cross-Zap.

Information is recorded on floppy disks using a magnetic recording head similar to a tape recorder. The information, called data, is recorded in circular rings called tracks. Each track can hold up to 6250 bytes (standard 5.25" floppy disk, single sided, double density).
However, because of control information only 4608 of these bytes are available for user programs and data. These 4608 are split into 18 groups of 256 bytes, each group being called a sector. Some of the control information is used to label each sector so the operating system can find it's unique label called an Address Mark. Preceding each sector one special byte called the Data Mark is recorded. Normally this is FB hex but, on TRS-80 disks can be F8 hex on some sectors.

All TRS-80 formats are based on the original TRSDOS developed for the TRS-80 Model I. This was a 35 track single sided single density format. The directory was in the middle of the disk at track 17. The directory track sectors used a different address (0FAH)mark to all others (0FBH). When double density formats were introduced the same principle was followed except that the directory track address mark used 0F8H for the address mark. The exception is Model III TRSDOS 1.3 where the marks are reversed: All sectors EXCEPT the directory have an F8 data mark while the directory sectors have an FB.

Typical TRS-80 disks have 35, 40 or 80 tracks, and may be single or double density, single or double sided. We will only discuss double density here since these are the only formats accessible to PC Cross-Zap.

5.1 The Boot Sector.

The first sector of the first track of all TRS-80 disks is used for the Boot Sector. This would normally contain a loader program so that the TRS-80 can get the Operating System into memory. The TRS-80 ROM contains a routine that loads this sector into memory at 4300H (4200H Model I) and then jumps to the start at 4300H. The first 2 or 3 bytes are set up so that the computer ignores them, typically they are

```
4300: 00          NOP       ; No Operation
4301: FE 11  CP   11H       ; Compare A with 17
```

The DOS loader can tell from the 3rd byte that the directory track is on track 17 (decimal). This is the only essential piece of information that must be in the Boot sector for it to be properly recognized by DOS. Two exceptions to the above may be found: TRSDOS 1.3 omits the NOP and DOS+ 3.4 sets most

significant bit of the byte containing the directory track to a 1 so it appears as 91H instead of 11H.

When DOS makes a data disk it puts a simple program in the boot sector that loads, clears the screen and pronounces itself 'Not a System DIsk' or 'This is a Data Disk'. However this is not necessary for any other purpose.

On Model I disks the boot sector is sector 0 and is in single density. Self booting disks must have at least sector 0 in single density so the most common solution for Model I double density disks is to make the whole of track 0 in single density and all the rest double density. As is mentioned elsewhere in this manual, the PC and hence PC Cross-Zap cannot read single density so you will not be able to access anything on track 0 of such disks.

5.2 The Directory.

The Directory of a disk provides a record of what files are on the disk, what space is used for each and where, and how much free space is left. On a TRS-80 disk, for each file there is an entry describing where the file is, it's size, date, passwords and other information.

For most types of TRS-80 operating system the directory occupies a complete physical track. In double density this means 18 sectors (36 for some double sided formats). The directory consists of 3 parts:

1) The GAT sector containing
   - A Granule Allocation Table (free space map).
   - A Track Granule Lockout Table.
   - The Disk Name, Date and Master Password.
   - The Disk AUTO Command.
   - Other Misc Information.

2) The HIT (Hash Index Table) sector containing
   - A quick means for DOS to determine whether a particular file is on the disk or not. Saves the DOS reading the entire directory.

3) The Directory proper. This occupies from 8 to 28 sectors, each sector is 256 bytes and can hold 8 file entries (excepting TRSDOS 1.3 which holds 5 entries). Each entry is 32 bytes (48 for TRSDOS 1.3).

A 32 byte directory entry contains the following information:

Byte	Use
00H	File Type and Access Level.
01H	Misc Flags and part of date.
02H	Rest of date (Day and Year)
03H	End Of File (EOF) byte.
04H	Logical Record Length
05-0CH	File Name - 8 characters
0D-0FH	File Extension - 3 chars.

```
            10-11H      Access Password
            12-13H      Update Password
            14-15H      No of Records in File.
            16-17H      Extent #1 descriptor
            18-19H      Extent #2 descriptor
            1A-1BH      Extent #3 descriptor
            1C-1DH      Extent #4 descriptor
All DOS types except TRSDOS 1.3:
            1E-1FH      Extended Entry flag & link pointer
TRSDOS 1.3 only:
            1E-2DH      Extent #5-#12 descriptors
            2E-2FH      Unused.
```

The Extent Descriptors define where the file is on the disk. It may be broken into several fragments or Extents. Each fragment is a whole number of grans in size and the Extent descriptor specifies where the extent starts and it's size in grans. If a file is broken into more than 4 fragments additional directory entries known as FXDEs (File eXtended Directory Entries) are used. The exception is TRSDOS 1.3 which uses 48 bytes per directory entry giving the possibility of 12 extents in one entry. FXDEs are not used in TRSDOS 1.3.

An FXDE looks just like a regular directory entry (FPDE, File Primary Directory Entry) except that bytes 02H through 14H are unused. Byte 01H is used as a reverse link pointer indicating where the previous FXDE or FPDE is to be found.

We will now discuss the meanings of the bytes in the Directory Entry in more detail and indicate the differences between DOSs if of any significance.

Byte 0H: File Type and Access Level

```
Bit
7       0=FPDE, 1=FXDE
6       0=User file, 1=System file
5       LDOS reserved, MultiDOS VFU % flag, others not
        used.
4       1=Active file, 0=Deleted (KILLed) file
3       0=Visible, 1=Invisible file
2 )     File Access Level:
1 )     000 Full      001  Kill   010 Rename 011 Not used
0 )     100 Write     101  Read   110 Exec     111 Access
```

Byte 01H Misc Flags and Month Date

```
Bits
7       Ø Shrinkable,  1 Non Shrinkable       )
        NewDOS: Ø File may expand             ) Not
6       Ø File backed up, 1 not backed up     ) used
        NewDOS: Deallocation flag             ) by
5       LDOS reserved, DOS+ not used          ) Multi
        NewDOS: Updated flag                  ) DOS )
```

```
4      LDOS only: MOD flag                    )      ) Not
3      )                                             ) used
2      ) Month              0000=no date             )  by
1      ) file              0001=January              ) NEWDOS
0      ) created           1100=December             )
```

Byte 02H: Rest of Date

```
Bit          All DOSs except        TRSDOS 1.3     NEWDOS

7      )                            )              )
6      ) Day  00=no date            ) Year         )
5      )  of  01=1st                ) offset       ) Not
4      ) Month      1F=31st         ) from         ) Used
3      )                            ) 1900         )
       )                            )              ) 00
2      ) Year 00=1980               ) 00=1900      )
1      ) from 07=1987               ) 56=1986      )
0      ) 1980                       )              )
```

Byte 03H EOF End of File

This byte indicates where, in the last sector of the file, the file ends. For example, 00 means the file ends exactly at the end of the last sector, 47 means that the first 47H bytes only of the last sector are meaningful. Bytes 48H to FFH do not belong to the file.

Byte 04H LRL Logical Record Length.

This byte tells DOS what size each record is for Random file access. For instance, if you have a mailing list program where each record is 80 (decimal) characters, the LRL will be 50H. This is used only by the TRS-80 type DOS to handle reading and writing one record at a time. Physically on the disk, the records are stacked end to end in a continuous stream. PC Cross-Zap does not need or use this byte in transferring files. Note that other DOSs such as MSDOS have no way of indicating in the directory what the LRL of a file is.

NewDOS 80 does not use this byte, except when the file is created. It is included for compatibility only.

Bytes 05 to 0FH File Name and Extension.

The first 8 bytes contain the file name, left justified and padded with spaces. The next 3 contain the filename extension, left justified and padded with spaces. Only the characters A-Z and 0-9 are permitted and the first character of both name and extension must be a letter. The extension can be omitted in which case it will appear as all spaces.

Bytes 10-11H File Update Password.

These 2 bytes represent the encrypted password that permits users to update files. The code for no password is:

TRSDOS 1.3:	EF5C
TRSDOS 2.7/8:	CD40
All others:	9642

To remove password replace with the correct code above.

Bytes 12-13H File Access Password

As above but this password, depending on the access code in Byte 00, restricts use of the file. Same codes as the Update Password above.

Note PC Cross-Zap ignores passwords for file transfer.

Bytes 14-15H Number of Records in File.

This field indicates the last sector written in the file expressed Least significant byte first. If EOF (Byte 03) is not zero then add 1 to number of records to get the total.

All DOSs count the number of records from zero except TRSDOS 1.3 and 2.7/8 which count from 1. Thus 0C00 means 12 sectors for all DOSs except TRSDOS 1.3 and 2.7/8 where it would mean 11 sectors.

Bytes 16-17H The 1st Extent Descriptor.

Byte 16H — contains the starting cylinder number (or lump number for NewDOS). If this byte is FF then this extent is not used.

Byte 17H — Bits 0-4 represent the number of contiguous grans in this fragment of the file. Note that 0=1 gran, 1=2 grans etc. EXCEPTION (as usual), TRSDOS 1.3 and 2.7/8 where the count is from 0. E.g. 1=1 gran, 2=2 grans etc.

Byte 17H — Bits 5-7 indicate the number of the 1st granule on the track or lump where this fragment of the file starts. Offset from 0, i.e. 0=starts on 1st gran, 1=2nd gran etc.

Example: 0322 means the file starts on Track 3, granule 3 and this fragment is 5 grans in size. (Not TRSDOS 1.3 or 2.7/8)

A Note about Relative Sectoring.

Here we come to a major difference between NewDOS 80 (also Model I DoubleDOS) and all other DOSs. NewDOS 80 uses logical units called LUMPS instead of physical tracks. A lump is normally 10 sectors = 2 grans and you must calculate the true physical position from the beginning of the disk (track 0). For instance the directory of NewDOS almost always starts on Lump

17 (decimal) that is 170 sector from the beginning of the disk. If the disk is a mixed density disk then ignore track 0 and start counting from the beginning of track 1. Thus the directory of a single sided, double density, Model I NewDOS disk will actually start on physical track 10, sector 8. (9*18+8 = 170).

In all other respects NewDOS uses the Extent Descriptor in the same way as other DOSs. It is only necessary to convert the lump number in Byte 16 to an equivalent physical track number and sector and then count forward by the number of grans indicated by bits 5-7 of Byte 17H to find the start of the extent on the disk.

Bytes 18-19H The 2nd Extent Descriptor.

As for 16-17H except if Byte 18H=FFH then no more extents.

Bytes 1A-1BH The 3rd Extent Descriptor.

As for 18-19H except if Byte 1AH=FFH then no more extents.

Bytes 1C-1DH The 4th Extent Descriptor.

As for 1C-1BH except if Byte 1CH=FFH then no more extents.

Bytes 1E-1FH FXDE pointer (Not TRSDOS 1.3)

If Byte 1EH is FFH then no more extents. Extent 4 was the last.

If Byte 1EH is FEH then at least one FXDE is present with extent descriptors describing more fragments of the file. In this case Byte 1FH points to the sector and entry containing the FXDE. Bits 3-7 give the directory sector number, and bits 0-2 give the number of the entry in the sector where the FXDE will be found. Note DOS tries to put the FXDEs in the same sector as the FPDE wherever possible.

Bytes 1E-2DH Extents 5 to 12 for TRSDOS 1.3 only

TRSDOS 1.3 does not use FXDEs, instead it uses 48 bytes for each file entry allowing these additional 8 Extent descriptors making a maximum of 12 in all. Bytes 2E-2FH are not used.

5.3 The Granule Allocation Table (GAT Sector)

GAT bytes 00-BFH

The principle use of the Granule Allocation Table is to provide a map of disk use, Granule by Granule. This map helps DOS quickly determine what

space is used and what is available when a new file is being added to the disk.

In fact the GAT proper contains no unique information that could not be obtained from the directory sectors. However to get that information DOS would have to read the entire directory which would be time consuming or would use a considerable amount of memory. PC Cross-Zap does this when it reconstructs the GAT by reading all directory sectors in the GAT Repair feature of Disk Navigator.

The GAT uses one byte for each track (or lump with NewDOS). On all DOSs that allow double sided operation (except NewDOS) a track includes all the sectors on the front and back. Each bit of each byte represents the status of one of the grans in the track or lump. Bit 0 for the 1st gran, bit 1 for the 2nd and so on. In some cases where there are only 3 grans per track then only 3 bits are used. If the gran is used then the bit is set to a 1, and if not used then it is a 0. All unused bits are set to 1.

With a few exceptions the GAT table occupies Bytes 00-5FH of the GAT sector. A similar table is kept at 60-BFH and this is used as a granule lockout table to disable access to sections on the disk that were found to be bad at format time. Bits are set to a 1 if the granule is locked out.

Where the disk has too many tracks or lumps to be represented in 60H bytes then the lockout table is not used. An example of this is NewDOS 80 double sided which has 144 lumps (on a 40 track ds disk) and uses bytes 00-8FH for the GAT.

GAT bytes C0-CFH Miscellaneous.

Bytes C0-CA. Not used except TRSDOS 2.7/8 puts an ID code of 27H at C0. If the DOS supports a hard disk it may use all the bytes from 00 to CA for the Granule Allocation Table (202 tracks max).

Byte CB Contains a code which indicates the type of DOS. 00 TRSDOS 2.7/8, 16/17 MultiDOS, 21 DoubleDOS, 32 TRSDOS 1.3, 34/35 DosPlus, 51 LDOS, 62 TRSDOS 6.x, 82 NewDOS 80

Byte CC No of tracks on disk in excess of 35. Exception: MultiDOS where it is number of logical cylinders. Not used by NewDOS.

Byte CDH LDOS: bits 0-2 Grans/Cyl-1, bit 5=1 if 2 sided, bit 6=1 id DDEN.
DOS+ 3.5 Bit 5 only, =1 if 2 sided disk.
MultiDOS: number of physical cylinders.
NewDOS, TRSDOS = 0

Bytes CE-CFH Disk Master Password:
When the disk master password is "PASSWORD"

Password code is:	TRSDOS 1.3:	D38F
	TRSDOS 2.7/8:	13CC
	All others:	E042

GAT bytes D0-DFH Disk Name and Date.

The first 8 bytes are used for the disk name. Normally limited to printable text characters.

The 2nd 8 bytes are used for the disk creation date in the format MM/DD/YY

GAT bytes E0-FFH DOS AUTO Command

32 bytes are available for a DOS automatic startup command which gets executed immediately DOS has finished booting. If the 1st byte at E0H is an 0DH then there is no AUTO command.

5.4 The Hash Index Table (HIT Sector)

The HIT sector of a TRS-80 Disk provides a quick way for DOS to determine if a file exists and if so, which directory sector it is in. This design helps to reduce the amount of memory needed by directory operations at the same time allowing fast and efficient disk access.

Each location in the HIT sector's 256 bytes corresponds to a possible directory entry. This imposes a maximum number of 256 files in any TRS80 directory no matter how big the disk or how many sectors are allocated to the directory. The location in the HIT is mapped from the directory sector number and entry according to one of three possible formulae:

Let S = Directory Sector Number (0 .. 27 max)
E = Entry number within sector (0 .. 7 max)

Then for	TRSDOS 1.3:	H = E+5S
	TRSDOS 2.7/8:	H = E+8S
	All others:	H = S+32E

H is the address of the byte in the HIT table.

The value of the byte, the Hashcode, is computed from the corresponding filename according to the following simple algorithm:

```
Start with hashcode byte=0
Loop 11 times doing the following:
        Get next character of filename
        Exclusive Or into Hashcode
        Rotate Hashcode byte circularly 1 bit left
        Loop till done
If result=0 then set Hashcode=1
```

If there is no filename or the file was deleted then the Hashcode byte is set to 0. File Extended Directory Entries (FXDE) take on the same Hashcode as the Primary Entry (FPDE).

The hashcode algorithm does not always give a unique result. Several different names can give rise to the same code. When this happens it is

called a hash collision. When DOS tries to find a file it computes the hashcode and searches the HIT table. If it finds a match it checks the corresponding directory sector. If the desired filename is not present because of a hash collision, it goes back to the HIT and searches for any further occurrences of the hashcode until it has either found the file or exhausted the HIT table.

Some locations in the HIT table are never used because of the way directory sector entries are mapped. NEWDOS 80 uses an unused HIT byte at location 1FH to store a number which indicates how many sectors (in excess of 10) are used in DIR/SYS which comprises the GAT, HIT and directory sectors. Legal values are 0, 5, 10, 15 and 20 (decimal).

==========

5.5 TRS-80 Disk Formats Summary

Model	DOS	Tracks	Sides	Track Ø Den	DIR DAM	1st Sector	Sector skew	Dir Track	Grans /Lump
I	DOS+ 3.4	40	1	S	F8	Ø	6	17	
I	DoubleDOS	40	1	S	F8	Ø	6	1Ø	2
I	LDOS/Sole	40	1	S	F8	Ø	3	17	
I	NewDOS 8Ø	40	1	S	F8	Ø	6	1Ø	2
I	NewDOS 8Ø	40	1	S	F8	Ø	3	5	2
I	NewDOS 8Ø	8Ø	2	S	F8	Ø	6	39	8
I	TRSDOS 2.8	40	1	S	F8	1	3	17	
I	MultiDOS	40	1	S	F8	Ø	3	17	
III	DOS+ 3.4	40	1	D	F8	Ø	6	2Ø	
I/III	DOS+ 3.5	40	1	D	F8	Ø	3	2Ø	
I/III	DOS+ 3.5	40	2	D	F8	Ø	6	2Ø	
I/III	DOS+ 3.5	8Ø	2	D	F8	Ø	6	4Ø	
I/III	LDOS 5.1	40	1	D	F8	Ø	3	2Ø	
I/III	LDOS 5.1	40	2	D	F8	Ø	6	2Ø	
I/III	LDOS 5.1	8Ø	2	D	F8	Ø	6	4Ø	
III	NewDOS 8Ø	40	1	D	F8	Ø	3	9	2
III	NewDOS 8Ø	40	2	D	F8	Ø	3	4	2
III	NewDOS 8Ø	8Ø	2	D	F8	Ø	3	38	8
III/4	MultiDOS 2	40	1	D	F8	Ø	3	17	
III/4	MultiDOS 2	40	2	D	F8	Ø	3	17	
III/4	MultiDOS 2	8Ø	2	D	F8	Ø	3	34	
III	TRSDOS 1.3	40	1	D	FB	1	3	2Ø	
4	DOS+ 4.Ø	40	1	D	F8	Ø	3	17	
4	DOS+ 4.Ø	40	2	D	F8	Ø	6	2Ø	
4	DOS+ 4.Ø	8Ø	2	D	F8	Ø	6	4Ø	
4	TRSDOS 6.2	40	1	D	F8	Ø	3	2Ø	
4	TRSDOS 6.2	40	2	D	F8	Ø	6	2Ø	
4	TRSDOS 6.2	8Ø	2	D	F8	Ø	6	4Ø	

Notes: The NewDOS 80 formats above have the following PDRIVE parameters:

```
M1 40tk SS DD:  TC=39 SPT=18 GPL=2 DDSL=17 DDGA=2
M1 40tk DS DD:  TC=39 SPT=36 GPL=2 DDSL=17 DDGA=6
M1 80tk DS DD:  TC=79 SPT=36 GPL=8 DDSL=35 DDGA=6

M3 40tk SS DD:  TC=40 SPT=18 GPL=2 DDSL=17 DDGA=2
M3 40tk DS DD:  TC=40 SPT=36 GPL=2 DDSL=17 DDGA=6
M3 80tk DS DD:  TC=80 SPT=36 GPL=8 DDSL=35 DDGA=6
```

When selecting a NewDOS 80 format you can change the number of tracks, but do not change the directory track.

TRS-80 Disk Formats Summary (Contd.)

Model	DOS	Tks	Sds	GAT ID	Sectors / Gran	Directory Start	Directory End	Passwords Disk	Passwords File
I	DOS+ 3.4	40	1	34	6	306	323	E042	9642
I	DoubleDOS	40	1	21	5	188	197	9642	9642
I	LDOS/Sole	40	1	51	6	306	323	E042	9642
I	NewDOS 80	40	1	82	5	188	197	E042	9642
I	NewDOS 80	40	2	82	5	206	223	E042	9642
I	NewDOS 80	80	2	82	5	1436	1465	E042	9642
I	TRSDOS 2.8	40	1	27	3	306	323	13CC	CD40
I	MultiDOS	40	1	17	6	306	323	9642	9642
III	DOS+ 3.4	40	1	34	6	360	377	E042	9642
I/III	DOS+ 3.5	40	1	35	6	360	377	E042	9642
I/III	DOS+ 3.5	40	2	35	6	720	737	E042	9642
I/III	DOS+ 3.5	80	2	35	6	1440	1457	E042	9642
I/III	LDOS 5.1	40	1	51	6	360	377	E042	9642
I/III	LDOS 5.1	40	2	51	6	720	755	E042	9642
I/III	LDOS 5.1	80	2	51	6	1440	1475	E042	9642
III	NewDOS 80	40	1	82	5	170	179	E042	9642
III	NewDOS 80	40	2	82	5	170	199	E042	9642
III	NewDOS 80	80	2	82	5	1400	1429	E042	9642
III/4	MultiDOS 2	40	1	FF	6	306	323	E042	9642
III/4	MultiDOS 2	40	2	FF	6	612	647	E042	9642
III/4	MultiDOS 2	80	2	FF	6	1224	1259	E042	9642
III	TRSDOS 1.3	40	1	32	3	306	323	D38F	EF5C
4	DOS+ 4.0	40	1	60	6	360	377	E042	9642
4	DOS+ 4.0	40	2	60	6	720	755	E042	9642
4	DOS+ 4.0	80	2	60	6	1440	1475	E042	9642
4	TRSDOS 6.2	40	1	62	6	360	377	E042	9642
4	TRSDOS 6.2	40	2	62	6	720	755	E042	9642
4	TRSDOS 6.2	80	2	62	6	1440	1475	E042	9642

Note: Directory start and end given as relative sector numbers counted from track 0, side 0, sector 0.

6.0 PC/MS-DOS Disk Formats

This chapter gives details of how data is organized and recorded on MS-DOS type disks. A knowledge of this information is essential if you are going to use the Disk Navigator feature of PC Cross-Zap.

Throughout this manual, reference is made to MS-DOS. This is the version of DOS sold with all PC-Compatible Machines such as the Tandy range. It was written by Microsoft. The version sold by IBM under the name PC-DOS is functionally identical and, as far as disk format considerations are concerned, it is exactly the same so wherever you read 'MS-DOS' you can be assured that the same discussion applies for PC-DOS.

6.1 MS-DOS Format Overview

We will restrict our discussion here to formats found on standard 3.5" and 5.25" standard (not high-density) floppy disks. The format has four parts:

The Boot Sector. In regular formats this is always the first sector (sector 1) of the first track (track 0). It serves to provide a loader for loading the operating system into memory. It may contain a table which defines the format parameters.

The FAT. Or File Allocation Table provides a map of disk space used by files and also a link list of clusters assigned to them. Two copies are kept. Normally the first FAT follows immediately after the boot sector and the second copy is contiguous with this.

The Directory. Usually follows immediately after the 2nd FAT and consists of at least 4 sectors. Each sector is divided into 32 byte entries, one for each file name. Each entry has the filename, attributes, date and time of creation, file size and a pointer to the starting cluster of the file. Looking at the corresponding position in the FAT gives the next cluster which in turn gives the next and so on.

The File storage area. The rest of available disk space is used to store files. The disks is filled from track 0 upwards. For double sided disks, side 0 is used first, then side 1, before going to the next higher track number.

The file storage area is divided up into sectors, as physically recorded on the disk these are usually 512 bytes in size. DOS manages them in terms of units called "Clusters". Typically, on single sided disks, a cluster is 1 sector or 512 bytes. On double sided disks a cluster is usually 2 sectors or 1024 bytes. On hard disks a cluster may be as large as 8192 bytes. The first cluster used, follows immediately after the directory and is cluster number 2. Subsequent clusters are numbered sequentially from this.

6.2 The Directory.

The Directory of an MSDOS disk consists of a number of 512 byte sectors divided into 32 byte sections. Each section describes one file on the disk. We will now discuss the meaning and use of each of the 32 bytes.

Byte(s)	Meaning and Use
0-7	Main Body of Filename, 8 characters Max, left justified and padded with spaces. If entry not previously used then Byte 0 = 0. If entry deleted then Byte 0 = E5H
8-10	Extension part of name, 3 characters maximum, left justified and padded with spaces.
11	File attribute byte. This defines the nature and use of the file. Meanings of the bits:

Bit 0 If 1 then file is Read Only
1 If 1 then file is Invisible
2 If 1 then file is a System File
3 If 1 then this is The Volume Name
4 If 1 then file is a Sub-Directory
5 If 1 then file has not been archived
6 Not defined
7 Not defined

12-21 Unused. Reserved for future use.

22-23 Time of file creation, Byte 22 bits 0-7, Byte 23 bits 8-15. Hours = bits 11-15, Minutes = bits 5-10, Seconds = bits 0-4.

24-25 Date of file creation, Byte 24 bits 0-7, Byte 25 bits 8-15. Year=1980 + bits 9-15, Month = bits 5-8, Day = bits 0-4.

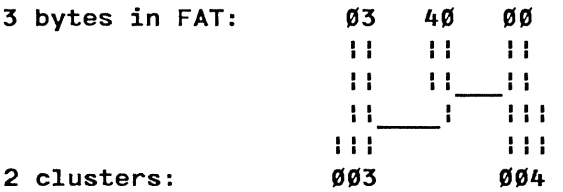
26-27 Starting Cluster in the file. Byte 26 is the least significant byte, 27 is most.

28-31 Files size in bytes. 28 is least significant byte, 31 is most significant.

6.3 The File Allocation Table.

The File Allocation Table or FAT provides a map of disk use, how much disk space is free and a linked list of pointers showing each cluster of each file is located.

The FAT is held on the disk as a table of 12 bit cluster pointers, one for each possible cluster on the disk. Looking at a typical FAT with PC Cross-Zap you will see a somewhat confusing situation. This is because each successive group of 3 bytes is used to hold two 12-bit cluster pointers in a scrambled fashion. Looking at a typical example you might see: 03 40 00. This must be decoded as follows: take the middle byte and split it into two nibbles, the 4 and the 0. Take the 2nd nibble (0 in this case) and put it on the front of the 1st byte to make 003. Take the 1st nibble (4) and put it on the end of the 2nd byte to make 004. Diagrammatically this can be represented as



6.4 An Example

Let us take a more complete example, the first part of an actual FAT display on a double sided MS-DOS 2.11 disk looked like this:

```
Address                        FAT  Bytes
-------------------------------------------------------------
0000          FD FF FF 03 40 00 05 60 00 07 80 00 09 A0 00 0B
0010          C0 00 0D E0 00 0F 00 01 11 20 01 13 40 01 15 60
0020          01 17 80 01 19 A0 01 1B C0 01 1D E0 01 1F 00 02
0030          21 20 02 23 40 02 25 60 02 27 80 02 29 A0 02 2B
0040          C0 02 2D F0 FF FF 0F 03 31 20 03 FF 4F 03 35 60
0050          03 37 80 03 ......
```

To make it readable, first we group it into threes:

```
-------------------------------------------------------------
        FD FF FF ¦ Ø3 4Ø ØØ ¦ Ø5 6Ø ØØ ¦ Ø7 8Ø ØØ ¦ Ø9 AØ ØØ
        ØB CØ ØØ ¦ ØD EØ ØØ ¦ ØF ØØ Ø1 ¦ 11 2Ø Ø1 ¦ 13 4Ø Ø1
        15 6Ø Ø1 ¦ 17 8Ø Ø1 ¦ 19 AØ Ø1 ¦ 1B CØ Ø1 ¦ 1D EØ Ø1
        1F ØØ Ø2 ¦ 21 2Ø Ø2 ¦ 23 4Ø Ø2 ¦ 25 6Ø Ø2 ¦ 27 8Ø Ø2
        29 AØ Ø2 ¦ 2B CØ Ø2 ¦ 2D FØ FF ¦ FF ØF Ø3 ¦ 31 2Ø Ø3
        FF 4F Ø3 ¦ 35 6Ø Ø3 ¦ ......
```

Then we rearrange the 2 halves of each middle byte:

```
Cluster              Decoded FAT table contents
Address
-------------------------------------------------
0000          FFD  FFF ¦ 003  004 ¦ 005  006 ¦ 007  008
0008          009  00A ¦ 00B  00C ¦ 00D  00E ¦ 00F  010
0010          011  012 ¦ 013  014 ¦ 015  016 ¦ 017  018
0018          019  01A ¦ 01B  01C ¦ 01D  01E ¦ 01F  020
0020          021  022 ¦ 023  024 ¦ 025  026 ¦ 027  028
0028          029  02A ¦ 02B  02C ¦ 02D  FFF ¦ FFF  030
0030          031  032 ¦ FFF  034 ¦ 035  036 ¦ .....
```

At each cluster address in the table is a 3 digit Hex (12 bit binary) number. If the number is non zero then the cluster is used or otherwise reserved. If the number is from FF8H to FFFH then this cluster is the last in the file. If it is from FF0H to FF7H then the cluster is bad and locked out. Otherwise the number indicates the next cluster in the file. The first 2 entries are never used except as a DOS type identifier. For standard 40 track floppy disks the following is true:

Ist byte in FAT	Disk type
FC	Single sided 9 sectors/track
FD	Double sided 9 sectors/track
FE	SIngle sided 8 sectors/track
FF	Double sided 8 sectors/track

The corresponding directory entry for the FAT example above looked like this:

```
Address                        Data
-------------------------------------------------------------
0000  4B45 524D 4954 2020 4558 4520 0000 0000  KERMIT  EXE
0010  0000 0000 0000 1180 480B 0200 00AF 0000  ............
0020  4D53 4B55 524D 4954 494E 4920 0000 0000  MSKERMITINI..
0030  0000 0000 0000 326F 310B 2E00 1001 0000  ............
0040  .......
```

So, on this disk the first file is KERMIT.EXE, its size is 0000AF00H (=44800 decimal) bytes and it's first cluster is number 0002 (from bytes 26,27 = 001AH,001BH above).

Referring back to the descrambled FAT at location 2 we see the number 003. This means the next cluster is number 003. Looking at location 003 we see 004 which means cluster 004 is next and so on. If we follow through to the end we see that at location 02C is 02D and at 02D is FFF which means cluster 02D is the last cluster in KERMIT.EXE. So our file occupies clusters 002 to 02D inclusive or 44 clusters. Each cluster on this disk is 1024 bytes so the file occupies 45056 bytes of disk space. The end of file is part way through the last cluster.

It should be noted that DOS 3.0 and higher is capable of writing disks with 16 bit allocation in the FAT entries. This is only likely to be found on Hard Disks. With the standard 12 bit scheme, up to 4080 clusters can be specified which is more than adequate even for a 1.2 Meg AT High Density disk. However, with a large hard disk this forces large cluster sizes - for instance a 20 Megabyte disk would force the use of 8 kbyte clusters which is very inefficient with large numbers of small files. With 16 bit FATs the cluster size can be reduced to say 2K resulting in much less wasted disk space.